

Installazione lato Server

OS: Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-141-generic x86_64)

HW: 8GB Ram, 4vCPU, 50GB storage

NodeJS

In questa sezione andremo ad installare l'ultima versione stabile di nodeJS ed il pacchetto npm che servirà per l'installazione dei moduli necessari a Nodered.

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -  
apt-get install -y nodejs node-gyp  
npm install -g npm  
echo "NODE_PATH=/usr/lib/node_modules" | tee -a /etc/environment
```

Nodered

Installiamo nodered e ne personalizziamo il funzionamento in modo da memorizzare i flussi nella home dell'utente di default ovvero ubuntu.

```
sudo npm install -g --unsafe-perm node-red@0.19.5  
  
wget https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/nodered.service -O /lib/systemd/system/nodered.service  
  
wget https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/node-red-start -O /usr/bin/node-red-start  
  
wget https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/node-red-stop -O /usr/bin/node-red-stop  
  
chmod +x /usr/bin/node-red-st*
```

Modificare il file /lib/systemd/system/nodered.service e cambiare i seguenti parametri:

User = [UTENTE]

Group = [UTENTE]

WorkingDirectory = /home/[UTENTE]

Installazione flussi

```
cd /home/[UTENTE]
```

```
mkdir .node-red
```

```
cd .node-red
```

```
curl -O https://raw.githubusercontent.com/igcomsrl/iot\_eCare\_flows/master/settings.js
```

```
curl -O https://raw.githubusercontent.com/igcomsrl/iot\_eCare\_flows/master/package.json
```

```
curl -o flows_cred.json
```

```
https://raw.githubusercontent.com/igcomsrl/iot\_eCare\_flows/master/manage\_sensor\_data\_cred.json
```

```
curl -o flows.json
```

```
https://raw.githubusercontent.com/igcomsrl/iot\_eCare\_flows/master/manage\_sensor\_data.json
```

```
npm install
```

```
systemctl daemon-reload
```

```
systemctl enable nodered.service
```

Configurare credenziali accesso dashboard

Impostare le seguenti variabili d'ambiente:

DASHBOARD_USERNAME e DASHBOARD_PASSWORD

Le credenziali di default sono: ubuntu G10m1R0m3

Configurare endpoint centrale operativa

Creare la cartella /data

```
mkdir /data
```

```
cd /data
```

```
sudo nano endpoints.json
```

Scrivere quanto segue:

```
{  
    "endpoint": "[ENDPOINT: ad es. http://37.148.227.52/semprevicini]",  
    "token_username": "[USERNAME PER TOKEN AUTENTICAZIONE IGCOR]",  
    "token_password": "[PASSWORD PER TOKEN AUTENTICAZIONE IGCOR]",  
}
```

Avviare il servizio: `systemctl start nodered.service`

Installazione grafana e influxdb

```
cd /tmp  
wget https://dl.influxdata.com/influxdb/releases/influxdb\_1.7.2\_amd64.deb  
sudo dpkg -i influxdb_1.7.2_amd64.deb  
wget https://dl.grafana.com/oss/release/grafana\_5.4.2\_amd64.deb  
sudo dpkg -i grafana_5.4.2_amd64.deb  
systemctl enable grafana-server.service influxdb.service  
systemctl start influxdb grafana-server  
sudo update-rc.d grafana-server defaults
```

Per blindare l'accesso al database e renderlo accessibile solo internamente al server stesso è necessario specificare il bind-address come segue.

```
vim /etc/influxdb/influxdb.conf    bind-address = "127.0.0.1:8086"  
systemctl restart influxdb
```

Configurazione Grafana → Influxdb

Creazione database Entrare nella bash del server:

```
influx
```

create database giomi

Collegarsi alla pagina di grafana esposta dal server http://<SERVER_IP>:3000 (credenziali di default admin admin, verrà chiesto di cambiarle al primo login)

1. Andare su configuration -> Data Sources -> Add data source -> selezionare Influxdb
2. specificare l'url come segue: <http://127.0.0.1:8086>
3. specificare il nome del database specificato in precedenza: giomi con le relative credenziali (se specificate)

Importante: il campo 'Name' nella sezione 'Settings' deve essere impostato come 'giomi'.

4. Premere il tasto a forma di '+' e premere 'Import', incollare il contenuto del file dashboard.json nel campo 'Or Paste JSON'.

Importante: l'uid della dashboard deve essere impostato a: XFWJlkuiz

5. Importare la dashboard di tipo scripted eseguendo il seguente comando:

```
wget https://raw.githubusercontent.com/igcomsrl/iot_eCare_ai/master/scripted.js -O  
/usr/share/grafana/public/dashboards/scripted.js
```

6. Modificare il file /etc/grafana/grafana.ini e cercare la stringa [auth.anonymous] e cambiare la stringa enabled = false in enabled = true

Riavviare grafana:

```
sudo systemctl restart grafana
```

Python

Installare il pacchetto Python, si consiglia la versione 3.5 . Per installare Python 3.5 eseguire il seguente comando:

```
sudo apt update && sudo apt install python3.5
```

Inserire la password del proprio utente e proseguire con l'installazione.

Successivamente installare il supporto di Python 3.5 agli ambienti virtuali. Questi permettono una migliore gestione del versioning delle librerie. È possibile installarlo attraverso il seguente comando:

```
sudo apt install python3.5-venv
```

Librerie

Creare una cartella chiamata pyScripts all'interno della cartella /data

```
sudo mkdir /data && sudo mkdir /data/pyScripts && cd /data/pyScripts
```

Creare un ambiente virtuale Python ed attivarlo:

```
sudo python3.5 -m venv env && source env/bin/activate
```

Installare all'interno dell'ambiente appena creato le seguenti librerie python nella versione specificata:

- matplotlib 3.0.3 (libreria che consente la creazione di grafici statistici)
- numpy 1.16.4 (libreria per calcoli matematico/scientifici)
- pandas 0.24.2 (libreria per gestire una grande mole di dati in modo efficiente)
- scikit-learn 0.21.2 (libreria di machine learning e modelli statistici)
- scipy 1.3.0 (libreria per i calcoli scientifici)
- keras 2.2.5 (libreria utilizzata per la creazione di reti neurali ad alto livello)
- tensorflow 1.14.0 (motore per reti neurali)

```
sudo env/bin/pip install influxdb==5.2.2  
matplotlib==3.0.3 numpy==1.16.4 pandas==0.24.2  
scikit-learn==0.21.2 scipy==1.3.0 keras==2.2.5  
sklearn==0.0 tensorflow==1.14.0  
openpyxl==2.6.4
```

Applicazione

Con i seguenti comandi si andrà a scaricare e installare l'applicazione sviluppata

```
sudo git clone  
https://github.com/igcomsrl/iot_eCare_ai.git
```


Cambiare parametri database

È possibile modificare i parametri di connessione al database InfluxDB attraverso la modifica di variabili di ambiente:

INFLUX_IP_AI: ip database influx

INFLUX_PORT_AI: porta database influx

INFLUX_USER_AI: username database influx

INFLUX_PW_AI: password database influx

INFLUX_DB_AI: nome database influx